Leisure-Management-Web-App

Jakob Pirker, 1231394

6. Juni 2016

1 Motivation und Ziele

1.1 RESTful-Webservice

1.2 Erweiterbarkeit

1.3 Anpassbarkeit

2 Problemstellungen und Lösungsansätze

2.1 Backend: Spring

2.1.1 Datenbank Anbindung

2.1.2 Datenübertragung (De- und Serialisierung der Daten)

2.2 Frontend: Javascript/Angular JS

2.3 Überblick: Frontend und Backend Interaktion

3 Backend

3.1 Spring

3.2 Architektur

3.2.1 Controller

3.2.2 Services

3.2.3 Repositories

3.3 Datenbankanbindung

3.4 Frontend-Interface

4 Frontend

4.1 Angular JS

4.2 Tabs

4.3 Tabellen

4.4 Eingabeformen

5 Installation und Anmerkungen

```
square

@Entity
@JsonPropertyOrder({ "Nachname", "Vorname", "Adresse" })
public class Person {

private static final Logger log = LoggerFactory.getLogger(Person.class);

@Id
@GeneratedValue
@Column
@JsonIgnore
private Long id;

@Column
private String surname;

@Column
private String forename;

@OneToOne
//    @JoinColumn(name = "address_id")
private Address address;

public Person() {
this.surname = "";
this.forename = "";
}

public Long getId() {
return id;
}

public void setId(Long id) {
this.id = id;
}

@JsonProperty("Vorname")
public String getForename() {
return forename;
}
```

```

@JsonProperty("Vorname")
public void setForename(String forename) {
this.forename = forename;
}

@JsonProperty("Nachname")
public String getSurname() {
return surname;
}

@JsonProperty("Nachname")
public void setSurname(String surname) {
this.surname = surname;
}


public void setAddress(Address address) {
this.address = address;
}

//
   ---------------------------------------------------------------
// serial- and deserializing specific content
```